Generate Collection

L14: Entry 1 of 17 File: USPT May 21, 2002

DOCUMENT-IDENTIFIER: US 6390371 B1
TITLE: Method and system for displaying information uniformly on tethered and remote input devices

Abstract Text (1):
A system for displaying information with a uniform visual display appearance on a variety of computer display devices, regardless of varying display capabilities of the display devices and of whether the display devices are tethered to the computer system or are remote. The system acts as an intermediary between an application program and the display devices. The system first receives information from the application program that is to be displayed on one of the display devices. The system then determines the location and display capabilities of the display device, and generates appropriate instructions that will display the received information on the display device. These instructions are generated in such a manner as to compensate for the varying display capabilities of the display devices, thus allowing the information to appear with a uniform visual display appearance on any of the devices to which the information may be displayed. If the display device is tethered to the computer system via a physical connection, the system directly invokes the display interface to the display device. If the display device is instead a remote device that is not local and tethered, the system generates instructions to invoke the display interface to the remote device and transmits the instruction to the remote device using a communication interface. If the application program is responding to information received from a peripheral device, the system can select an appropriate display device on which to display the response information.

Application Filing Date (1):
19980213

Brief Summary Text (4):
In the past, personal computer systems were generally stand-alone systems in which the computer displayed information to a single user on a single display device (e.g., a computer monitor). However, as computers have become increasingly sophisticated, they frequently have a variety of auxiliary display devices, in addition to the primary display device, to which they can display information. For example, peripheral devices such as input/output (I/O) devices often have integrated displays that are used to display information about the peripheral device. If these peripheral devices can also display information that they receive from the computer system, then they can act as auxiliary display devices for the system. Examples of peripheral devices that often have such integrated displays include output devices such as printers and input devices such as scanners. The integrated displays of such devices can vary widely in their size and display capabilities, such as the number of display lines, the number of colors, the types and sizes of fonts, and the ability to display non-textual information such as images, animation, or full-motion video. In addition to these primary and auxiliary display devices that are part of the computer system (i.e., local display devices), computers increasingly can also display information on display devices that are part of other computer systems (i.e., non-local display devices) via a shared network between the systems. These non-local display devices can also vary in size and display capability.

Brief Summary Text (5):
Each display device typically has a display interface that allows a program executing on a computer system to invoke the display capabilities of the device. These display interfaces (e.g., a set of window display functions to display information in a window of a graphical user interface) can differ for different types of display devices, and even for different devices of the same type. For example, since operating systems normally provide a display interface to the primary display device, different operating systems typically provide different display interfaces. Thus, the display interface to a primary display device on a computer system running the WINDOWS operating system will be significantly different than the display interface to that same display device on a computer system running the MACOS operating system. In addition to the differences in display interfaces caused by operating systems, peripheral devices with integrated displays typically have proprietary display interfaces that are different from the display interface to the primary display device.

Brief Summary Text (6):
Many local display devices are tethered to the computer system via a physical connection such as a copper wire or a fiber-optic cable. However, in addition to local tethered display devices, a computer system may need to display information on display devices that are local but untethered (e.g., a wireless I/O device) or to display devices that are non-local (e.g., a primary display of another computer system). Devices that are non-local and devices that are local but untethered are referred to as remote devices since they often can be located some distance from the computer system. As described above, an application program typically invokes a function of the appropriate display interface to display information on a tethered local display device. However, to display information on a remote device, it is necessary to use a communication interface (e.g., a network message protocol) in addition to a display interface. Communication interfaces allow a program executing on a computer system to communicate information to a remote device over a variety of transmittal mediums such as copper wire, fiber-optic cable, or various wireless mediums. Therefore, to display information on a remote display device (e.g., a local wireless auxiliary display device or a non-local primary display device of another computer system), an application program will typically have to generate information that can invoke the display interface to the remote device, and will then have to invoke a function in a communication interface to transmit the information to the device. When the information is received by the remote device, the information can then be used to invoke the display interface. Typically, the information transmitted will be an instruction that invokes a function of the display interface.

Brief Summary Text (7):
Due to differences in display interfaces and display capabilities, separate application programs are typically developed when information must be displayed using different display interfaces. For example, a product like Microsoft Word will typically have one application program for a computer system running the WINDOWS operating system, and a separate application program for a computer system running the MACOS operating system. Even if the same information is to be displayed, different application programs are often created. For similar reasons, it is also typically the case that separate application programs are developed for different types of communication schemes (e.g., local tethered display versus local untethered display). Even if a single application program is developed that can display information on either a local tethered or a remote display device, the application program will have to contain separate code to invoke the display interface to the tethered device, to invoke the communication interface to the remote device, and to invoke the display interface to the remote device.

Brief Summary Text (8):
Having multiple separate application programs creates a variety of problems. For

example, the developers for each application program will have to determine the display interfaces and communication interfaces that the application program will use, and will then spend a large amount of effort to create the appropriate program code to display information using these interfaces. This effort will largely be duplicated for each of the application programs that are created. In addition to duplication of effort related to displaying information, application programs which display the same information on different display devices will also duplicate much of the code used to generate the information that will be displayed. Moreover, if separate application programs with similar functionality are modified over time, effort will have to be expended to modify each of the programs as well as to maintain consistency between the programs.

Brief Summary Text (9):
In addition to these problems of duplication of effort, it is often desirable that regardless of the display device on which information is to be displayed, that the information have a uniform visual display appearance (i.e., to appear substantially identical to users). For example, documentation will often have to be prepared for each different application program visual display appearance, and users will have to be trained on the use of each of these different visual display appearances. However, it is difficult to create a uniform visual display appearance on different display devices. Even if a single application program is displaying the same information on two display devices of the same type, the generated visual display appearances are likely to be different if different display interfaces are used. These differences in the visual display appearances can arise due to different functionality provided through the different display interfaces, or from the same type of functionality provided in a different manner (e.g., one display interface may use pop-up menus to display information to be selected by the user, while another display interface may use a pull-down menu or toolbar buttons). The problem of differences in visual display appearances is only exacerbated when there are separate application programs to display the same information, and when different display devices have different display capabilities. Even if an explicit effort is made to display the same information with a uniform visual display appearance, it is likely to be difficult and time-consuming to maintain this uniformity.

Brief Summary Text (10):
Other difficulties can arise when developing an application program that will display information on a display device other than the primary display device. For example, it may be difficult to debug problems with the application program due to the display interface or the communication interface being used (e.g., if an error occurs after interface functionality is invoked, only limited information may be returned from the interface about the error). Alternatively, if the display device for the program is a peripheral device with an integrated display, it may be difficult to obtain such a device for testing. Thus, various problems exist with displaying information on a variety of display devices, particularly when a uniform visual display appearance is desired.

Brief Summary Text (12):
Some embodiments of the present invention provide a method and system to display information with a uniform visual display appearance on a variety of computer display devices, regardless of varying display capabilities of the display devices and regardless of whether the display devices are tethered to the computer system or are remote from the computer system. The system acts as an intermediary between an application program and the display devices. The system first receives information from the application program that is to be displayed on one of the display devices. The system then determines the location and display capabilities of the display device, and generates appropriate instructions that will display the received information on the display device. These instructions are generated in such a manner as to compensate for the varying display capabilities of the display devices, thus allowing the information to appear with a uniform visual display appearance on any of the devices. If the display device is tethered to the computer

First Hit Fwd Refs

☐ Generate Collection

L14: Entry 2 of 17 File: USPT May 9, 2000

DOCUMENT-IDENTIFIER: US 6061064 A
TITLE: System and method for providing and using a computer user interface with a view space having discrete portions

Abstract Text (1):
A system and method associate each of a plurality of computer applications with a corresponding physical location external to the computer and display a given one of the applications when the user focuses attention on the physical location associated with that application. Preferably the display as a view window in a graphical user interface, and the user has means for moving that window relative to the given application. The computer can be a portable and display device can be head mounted. Preferably an input device enables the user to interact with the given application, and preferably the physical locations bring to mind their associated applications. In some embodiments, an identifier, such as a bar code or a coded transmitter, is placed near each of the physical locations to help detect when the user focuses attention on that particular location. The invention also provides a head mounted unit which projects a visual image to the user wearing it. The unit also includes an object

Application Filing Date (1):
19960827

Brief Summary Text (7):
The most common approach has been window based software. Windowing systems attempt to maximize the use of the screen space of a display terminal by providing overlapping windows and icons. The window operating environment, although useful, is often frustrating to operate. The user is required to spend an inordinate amount of time moving, resizing, and opening and closing various windows and icons on the display space. The opening and closing of a window is often slow. Overlapping windows can be aggravating to the eye. It is also difficult to manipulate information within windows. The physical size of the display terminal limits the size of each window, the number of windows that can be displayed at a given time, and, in the case of graphic intensive applications, is often too small to display an image of an object in its entirety.

Brief Summary Text (8):
Another approach to increasing the display surface area of a computer is to simply use a larger monitor. Several companies are marketing twenty-eight (28) inch diagonal monitors. These extra-large monitors do increase the display capabilities of the computer to some degree, but the problems outlined above are still present. These monitors are also prohibitively expensive to build and difficult to ship to customers. One such monitor currently on the market weighs over two hundred pounds and is more than thirty inches deep. This monitor is clearly impractical for standard desktop computers.

Brief Summary Text (17):
One aspect of the present invention involves a system and method to enable a user to interact between the real world and a virtual world created by a computer. It involves associating each of a plurality of computer applications with a corresponding physical location external to, and not physically connected to, the

computer. It further involves viewing, or displaying, a given one of the applications when the user focuses attention in the general direction of the physical location with which the given application has been associated.

Brief Summary Text (28):
According to another aspect of the invention a computational device is provided which comprises a user interface display, a display generator, and a motion sensor. The display generator generates a display object on the user interface display. The motion sensor is coupled to the display generator, and it directs the display object to move within the user interface display in response to motions sensed by the motion sensor. In a preferred embodiment, the computational device is a hand held computer and the display object is a cursor. In this embodiment, the cursor moves within the user interface display in response to hand motions of the user as detected by the motion sensor.

Detailed Description Text (4):
swiveling chair 12, a computer 14 mounted at the base of the chair 12, a platform 16 for supporting computer peripheral devices such as a keyboard 18 and a mouse 20, a head mounted display 22 and a position sensor 24 (housed inside computer 14), including a transmitter 26 mechanically coupled to the head mounted display 22 and a receiver 28 mechanically connected to a stationary reference point 30. The reference point can be located above the user's head as illustrated in FIG. 1, at the base of the chair 12, or any other stationary location in the vicinity of the video display system 10.

Detailed Description Text (5):
Referring to FIG. 2, the relationship between a frame buffer, a view port and a view window in a virtual view space in the video system 10 is illustrated. The virtual view space 40 is the total image area in the video display system 10. The virtual view space 40 is 360.degree. and has a height of 135.degree.. The virtual view space 40 is shaped like a "cylinder" which surrounds the user. In the preferred embodiment, the total size of the virtual view space 40 is equivalent to a wall size display having the dimensions of approximately eight (8) feet by three (3) feet.

Detailed Description Text (10):
During operation, the user may navigate the virtual view space 40 by rotating his or her head from side to side, tilting his or her head up or down, or swiveling in chair 12. The position sensor 24 permits the video display system 10 to emulate a video environment that has two degrees of freedom. The position sensor 24 generates rotation (yaw) information and vertical movement (pitch) information in response to the movement of the transmitter 26 on the head mounted display 22 with respect to the receiver 28 at the reference point 30. It should be noted that in the preferred embodiment, only yaw and pitch movement is measured. It is within the scope of the present invention to measure other motions, such as scaling (forward/backward), roll, lateral, side-to-side, and up/down.

Detailed Description Text (12):
To generate an image of the view port 34 in the virtual view space 40, computer 14 retrieves the pixel information from the frame buffer 42 that corresponds to the view port 34. The pixel information is stored in (1120.times.900) memory locations. The pixel information is subsequently transferred to the head mounted display 22. The pixel information displayed in the head mounted display 22 is referred to as view window 36. The view window 36 includes (1120.times.900) pixels and has the dimensions of (25.degree..times.20.degree.) within the virtual view space 40.

Detailed Description Text (13):
The view window 36 may include a horizontal scroll bar 37 which designates the yaw position and a vertical scroll bar 38 which designates the pitch position of the view window 36. In the example illustrated in FIG. 2, the display window 36 is

located at a position of approximately 270.degree. yaw and 70.degree. pitch in the virtual view space 40. The scroll bars help the user keep track of the current location in the virtual view space 40 and they help to locate regions in the virtual view space 40 not currently displayed in the view window 36.

Detailed Description Text (18):
Referring to FIG. 3, a block diagram of the video display system 10 is shown. The video display system 10 includes a main CPU 50, a system bus (SBus) 52, a graphics processor 54, a memory 56 including the frame buffer 42 and code space 57, an arbiter 58, a scan line generator 60, graphics output controller 62, first-in-first-out (FIFO) buffer 64, display timing generator 66, the position sensor 24 and the head mounted display 22.

Detailed Description Text (19):
The graphics processor 54, memory 56, arbiter 58, scan line generator 60, graphics output controller 62, FIFO 64 are all provided on a single buffer display card 70. The display timing generator 66 is an electronic card marketed by Reflection Technology along with the head mounted display 22. A ribbon cable (not shown) is used to couple the display buffer card 70 and the display timing generator card 66. The two cards are inserted in the housing of computer 14. The position sensor 24 is wired to the buffer display card 70 through the RS-232 port (not shown) of the computer 14.

Detailed Description Text (24):
Modifications to Open Windows were made to take advantage of and to extend the user interface capabilities of the virtual view system 10 environment. These modifications predominantly include changes to the window manager of Open Windows. For example, when the user opens a new window, the window manager ensures that the window is displayed in the current view window 36. Similarly, the window manager is modified so that dialog boxes appear in the current view window 36. When the user invokes a full screen or a full height function of an image, the window manager has been modified so that the image is resized to match that of the view window 36, and not the size of the frame buffer. The window manager is also modified to account for the fact that the cursor is primarily controlled by the position sensor 24. The window manager insures that the cursor always appears in the current view window 36. The mouse 20 only secondarily controls the cursor by determining its position within the current view window 36. Many of these modifications are desirable because the frame buffer 42 is much larger than the view window 36, which differs from than prior art video systems where the frame buffer and the display are the same size.

Detailed Description Text (70):
The head mounted unit 161 also includes other electronic component necessary to interface to or drive the display 162, the photo detectors 135, the microphone 166, the speaker 168, and the bar code reader 170, as will be understood by those skilled in the art of electronic design. It also includes wires 172 which communicate signals from the head mounted unit to a computer. Preferably this computer is small enough to be worn by the user, such as in a back pack, so the user is free to walk around while using the computer. Such a computer can be constructed from a SPARCstation computer, of the type described above, with the standard AC power supply replaced with a battery power supply. A computer with enough power to run Sun OS and the Open Windows graphical user interface could be built to fit within a user's pocket using today's technology. Within the coming decade such computers are expected to be small enough to fit within a head mounted unit such as that shown in FIG. 9.

Detailed Description Text (78):
Step 154 performs computation on a communication link associated with an application when user input or computation in steps 153 or 155 requires it. In computers with a graphical user interface the communication links are usually

windows created when an application is opened. Such windows communicate from their application to a user by providing visual information or other output. They communicate from the user to their application in response to user mouse clicks, mouse drags, or typing on objects shown in the window. In character based computers the communication links can be screens or non-graphic windows. In audio computers the communication links can be connections to audio input and output.

Detailed Description Text (90):
The bar code 184F is unusual in that it is worn as part of an identity badge on the clothing of a person 204, rather than being stuck on an inanimate object. The windows 180F placed in its associated desktop 182F are those which a user wishes to use when he or she sees the particular person wearing the bar code pattern 184F. One such window might include information about the person wearing the bar code, such as data out of a corporate personnel directory. Another might include matters which the user wishes to discussed with the person, and so on. Other desktops, associated with bar codes worn by other persons, can contain windows associated with such other persons.

Detailed Description Text (126):
Step 338 makes the desktop associated with the detected ID the selected desktop. It displays that desktop at its last displayed location and size, covering any other desktops in that portion of the display, showing any windows of the portion of the desktop displayed, and giving input/output focus to the window last selected in that display.

Detailed Description Text (140):
If the user drags on the title bar of the desktop, step 368 will move its position relative to the visual display. This position will be maintained next time the desktop is displayed again.

Detailed Description Text (167):
FIGS. 35 and 36 illustrate how the invention can be used not only to control the display of information on an individual's head mounted display, but also on a communal display 434 which acts as an electronic bulletin board. In this case the communal display is treated in the desktop outline just as is each user's individual display. For example, the display "Front Door Msg. Screen" shown in FIG. 29 is a communal display. As is indicated in FIG. 36, in which that display's part of the desktop outline has been expanded, it has the same type of outline structure as does the individual display associated with "Bruce".

Detailed Description Text (187):
If the program is in the command recognition mode, the mode in which it responds to user gestures to control desktop display and cursor functions, the test of step 494 will be met, causing steps 496 and 500 to be performed.

Detailed Description Text (201):
The measuring step 92 of FIG. 5, the position change calculating steps of FIG. 8, and the detecting steps 158 and 254A of FIG. 11 and of FIGS. 17A-17B, respectively, can all be performed in ways other than by measuring motion or position of the user's head. This includes detecting the position of a portable computer or visual display as shown in FIG. 37, detecting the position of the user's portable computer, such as by the intra-building and global positioning locators discussed with regard to FIGS. 15 and 30-33, or detecting the position of a part of the user's body, as shown in FIGS. 38 and 39.

Detailed Description Text (204):
The computation which equates changes in motion or position with changes in display can be varied in different embodiments of the invention. For example, the amount of head or other user movement required to navigate from one screen to the next in the view spaces of FIGS. 2, 6, and 7 can be varied greatly. In some situations it is

advantageous to make a small amount of user motion cause the transition from one screen to the next. In others it is desirable to allow the user a wide degree of movement without changing screens. For example, when very small amounts of motion are sufficient to cause screen changes, movement through a quantized space of the type shown in FIGS. 6 or 7 can be used to create a continuous "motion picture" image.

Detailed Description Text (209):
In alternate embodiments of the aspect of the invention, shown with regard to FIG. 11 and the figures that follow it, items other than desktops are associated with external locations. For example, in some such embodiments windows are associated directly with external locations, rather than being placed at desired locations in desktops associated with such locations. In other embodiments of this aspect of the invention computational objects other than GUI windows are associated with external locations, including character based windows, images, text messages, audio messages, OOP type objects, changes in input or output direction, different locations within a document, changes in modes of user interface operation, the turning of a display on or off, the zooming of a display's field of vision in or out, inputs in a command grammar, etc.

Detailed Description Text (210):
Alternate embodiment of the invention can operate with different types of hardware than that described above. As has been mentioned, different types of visual displays and motion or position sensing apparatus can be used. Computers other than those manufactured by Sun Microcomputer, Inc. described above can be used. For example, pocket sized or wrist mounted computers could be used. The computers could be multiprocessing computers or non-Von Neumann computers. Neural net or fuzzy logic computer could be used, for example, to automatically associate a communication link with a given location or object if the user often uses that link when pointing at or near that location or object.

Detailed Description Text (213):
The command interpreter shown in FIG. 16 used to control a head-mounted, see-through display could use input other than eye closings to control its operations. For example, in other embodiments eye ball direction, speech, head nods, or other body gestures could be used. Similarly the command gestures used with embodiments of the invention like that of FIGS. 38 and 39 which views the motion of body parts can differ. For example, in a system monitoring head position with a video camera, blinks or facial expressions can be used to simulate mouse clicks or indicate when pointing is to be interpreted as a command.

Other Reference Publication (5):
Deering, Michael, "Explorations of Display Interfaces for Virtual Reality", Virtual Reality, 1993 Symposium, pp. 141-147.

CLAIMS:

17. The system of claim 13, further comprising an input device to enable the user to interact with the graphical user interface windows shown in the visual display.

24. The method of claim 20, further comprising the step of enabling the user to interact with the graphical user interface windows shown in the visual display through an input device.

First Hit Fwd Refs

Generate Collection

L14: Entry 4 of 17 File: USPT Oct 26, 1999

DOCUMENT-IDENTIFIER: US 5973702 A
** See image for Certificate of Correction **
TITLE: Oriented view system having a common window manager for defining application window areas in a screen buffer and application specific view objects for writing into the screen buffer

Abstract Text (1):
An object-oriented view system controls the display of screen graphics for a plurality of application programs, each of which generates graphical information for display in a window assigned to it. The view system has a system window manager which is common to all of the application programs and which defines application window areas on the display screen and corresponding application window storage areas in the display screen buffer. Each application program instantiates a view system object from class information in the computer operating system. Each view system object includes a view object with program code that directly stores screen display information generated by the application into the screen buffer. This arrangement allows the application programs to avoid the conventional "bottleneck" that develops when all of the screen display information must be stored in the screen buffer by the common system window manager.

Application Filing Date (1):
19950607

Brief Summary Text (6):
This invention generally relates to improvements in computer systems and, more particularly, to operating system software for managing drawing areas, called views, inside of a window display area in a graphic user interface.

Brief Summary Text (10):
Each window region generally displays information which is generated by the associated application program and there may be several window regions simultaneously present on the desktop, each representing information generated by a different application program. An application program presents information to the user through each window by drawing or "painting" images, graphics or text within the window region. The user, in turn, communicates with the application by "pointing at" objects in the window region with a cursor which is controlled by a pointing device and manipulating or moving the objects and also by typing information into the keyboard. The window regions may also be moved around on the display screen and changed in size and appearance so that the user can arrange the desktop in a convenient manner.

Brief Summary Text (29):
The lowest or computer hardware level includes the basic computer and associated input and output devices including display monitors, keyboards, pointing devices, such as mice or trackballs, and other standard components, including printers and disc drives. The next or "component driver software" level consists of device-dependent software that generates the commands and signals necessary to operate the various hardware components. The resource control and communication layer interfaces with the component drivers and includes software routines which allocate resources, communicate between applications and multiplex communications generated

by the higher layers to the underlying layers. The view system handles the user interface to basic drawing operations, such as moving and resizing views, activating or inactivating views and redrawing and repainting views. The final user interface layer provides high level facilities that implement the various controls (buttons, sliders, boxes and other controls) that application programs use to develop a complete user interface.

Brief Summary Text (31):
Accordingly, it is an object of the present invention to provide a view system which can interface with application threads in such a manner that the screen display generated by each application thread can be quickly and effectively redrawn.

Brief Summary Text (34):
It is yet another object of the present invention to provide a view system which allows application developers who need detailed control over the screen display process to achieve this control by means of a full set of display control commands which are available, but need not be used by each application thread.

Brief Summary Text (35):
It is yet another object of the present invention to provide a view system which provides application developers with a powerful and flexible drawing environment which includes a virtual coordinate space, arbitrarily shaped views (and windows) and up-to-date drawing state information to facilitate rapid, accurate drawing from multiple threads of execution.

Drawing Description Text (6):
FIG. 4 is a schematic block diagram of a modified computer system showing the interaction between a plurality of application threads, the viewing framework, the window manager, and the screen buffer in order to display graphic information on the display monitor.

Detailed Description Text (14):
In the same way that an application framework provides the developer with prefab functionality for an application thread, a system framework, such as that included in a preferred embodiment, can provide a prefab functionality for system level services which developers can modify or override to create customized solutions, thereby avoiding the awkward procedural calls necessary with the prior art application frameworks programs. For example, consider a display framework which could provide the foundation for creating, deleting and manipulating windows to display information generated by an application thread. An application software developer who needed these capabilities would ordinarily have to write specific routines to provide them. To do this with a framework, the developer only needs to supply the characteristics and behavior of the finished display, while the framework provides the actual routines which perform the tasks.

Detailed Description Text (19):
The interaction of an application thread with the view system is illustrated in more detail in schematic diagram FIG. 5. As previously mentioned, the view system (illustrated as box 510 in FIG. 5) is an object-oriented program. Accordingly, an application thread 508 interfaces with the view system by creating and manipulating "objects". In particular, each application thread creates a view hierarchy object, for example, view hierarchy object 512 in order to communicate with view system 510. The application thread 508 then communicates with the view hierarchy object 512 by creating a view object 506 and installing it in the hierarchy as shown schematically by arrow 502. The view system itself is a collection of objects which is created when the application program is started. The view system 510 interfaces with the operating system 500 via a data stream 504 to perform window operations on behalf of the application program and view system 510.

Detailed Description Text (20):
As will hereinafter be described in more detail, each view object 506 includes a small data store or "cache" area, called the drawing state 514 which is used to store the associated view visible area and other drawing-related state (coordinate system etc.). When the application thread desires to redraw the information in one of its associated views, the view object first checks cache status. If the information stored in the cache has not been changed or invalidated, then this information is used to redraw the window. The use of the cache area reduces the time necessary to complete a redrawing operation.

Detailed Description Text (21):
Since many view objects may be created simultaneously in order to simultaneously display many views within a window, each view object 506 communicates with the view system 510 by means of multitask-safe method calls 502. The view system communicates with the operating system via data stream 504 by creating "stream" objects which contain the software commands necessary to transfer information from one object to another. For example, when operating system 500 desires to transfer information to view system object 510, operating system 500 creates a stream object which "streams" the data into view system object 510. Similarly, when view system object 510 desires to transfer information back to operating system 500, view system object 510 creates a stream object which "streams" the data into window object 500. Such stream objects are conventional in nature and not described in detail herein. The stream objects which carry data from operating system 500 to view system object 510 and the stream objects which carry information from view system object 510 to operating system 500 are illustrated collectively as arrow 504.

Detailed Description Text (31):
Although the controls discussed above generally cannot be moved or resized by the application thread, the content view and child views are usually totally under control of the application thread. When an application thread has several views, or several application threads, which are running simultaneously, and displaying information views, changes in the size or the position of one view will change the displayed or visible areas of views which are "under" the changed view. FIGS. 7A and 7B illustrate how a manipulation of one view associated with an application can change the visible areas of other views that are associated with the same application and inside the same window.

CLAIMS:

1. An apparatus for controlling a display device to generate a display having a plurality of windows displayed on a desktop background, each of the plurality of windows being assigned to, and displaying screen information generated by, one of a plurality of application programs, the apparatus comprising:

a screen buffer for storing screen information generated by the plurality of application programs and displaying stored information on the display device;

a view system monitor responsive to screen information generated by the plurality of application programs for dividing the screen buffer into a plurality of storage areas, each of the plurality of storage areas storing the screen information for one of the plurality of windows; and

a plurality of view system objects, each of the plurality of view system objects being part of one of the plurality of application programs and operating to store screen information directly in one of the screen buffer storage areas associated with a window assigned to the one application program.

4. An apparatus according to claim 1 wherein the apparatus comprises an object-oriented operating system for controlling the display device and each of the

plurality of view system objects is created by one of the plurality of application programs using class information located in the operating system.

7. An apparatus according to claim 1 wherein each of the view system objects comprises:

a plurality of view objects, each of the plurality of view objects comprising commands for storing screen information into the screen buffer for one window associated with the view system object; and

means controlled by one of the plurality of application programs for arranging the plurality of view objects into a view hierarchy.

9. An apparatus according to claim 1 wherein the apparatus comprises an object-oriented operating system for controlling the display device and the view system monitor is located in the object-oriented operating system.

10. A method for controlling a computer system having a display device to generate a display having a plurality of windows displayed on a desktop background, each of the plurality of windows being assigned to, and displaying screen information generated by, one of a plurality of application programs, the method comprising the steps of:

A. storing screen information generated by the plurality of application programs in a screen buffer;

B. displaying stored information on the display device;

C. dividing the screen buffer into a plurality of storage areas, each of the plurality of storage areas storing the screen information for one of the plurality of windows using a common view system monitor;

D. creating a plurality of view system objects, each of the plurality of view system objects being part of one of the plurality of application programs; and

E. each of the plurality of view system objects storing screen information directly in a storage area associated with a window assigned to the application program of which the view system object is a part.

12. A method according to claim 11 wherein the computer system comprises an object-oriented operating system for controlling the display device and wherein step D1 comprises the step of:

D1A. creating each of the plurality of view system objects using class information located in the operating system.

16. A method according to claim 10 wherein step D comprises the steps of:

D3. for each of the view system objects, creating a plurality of view objects, each of the plurality of view objects comprising commands for storing screen information into the screen buffer for one window associated with the view system object; and

D4. arranging the plurality of view objects into a view hierarchy.

17. A method according to claim 10 wherein the computer system comprises an object-oriented operating system for controlling the display device and wherein step C comprises the step of:

C1. creating the view system monitor in the object-oriented operating system.

18. A computer program product for controlling a computer system having a display device, a screen buffer and a mechanism for displaying information stored in the screen buffer on the display device to generate a display having a plurality of windows displayed on a desktop background, each of the plurality of windows being assigned to, and displaying screen information generated by, one of a plurality of application programs, the computer program product comprising a computer usable medium having computer readable program code thereon including:

program code for generating a common view system monitor to divide the screen buffer into a plurality of storage areas, each of the plurality of storage areas storing the screen information for one of the plurality of windows;

program code for creating a plurality of view system objects, each of the plurality of view system objects being part of one of the plurality of application programs; and

program code in each of the plurality of view system objects for storing screen information directly into a storage area associated with a window assigned to the application program of which the view system object is a part.

20. A computer program product according to claim 19 wherein the computer system comprises an object-oriented operating system for controlling the display device and wherein the program code for controlling the plurality of application programs comprises program code for creating each of the plurality of view system objects using class information located in the operating system.

24. A computer program product according to claim 18 wherein the program code for creating at least one view object in each of the view system objects comprises:

program code in each of the view system objects for creating a plurality of view objects, each of the plurality of view objects comprising commands for storing screen information into the screen buffer for one window associated with the view system object; and

program code for arranging the plurality of view objects into a view hierarchy.

25. A computer program product according to claim 18 wherein the computer system comprises an object-oriented operating system for controlling the display device and wherein the program code for generating a common view system monitor comprises:

program code for creating the view system monitor object in the object-oriented operating system.

First Hit Fwd Refs

Generate Collection

L14: Entry 10 of 17 File: USPT Mar 24, 1998

DOCUMENT-IDENTIFIER: US 5731805 A
TITLE: Method and apparatus for eyetrack-driven text enlargement

Abstract Text (1):
Apparatus, methods, systems and computer program products are disclosed that automatically expands the computer displayed information that interests a computer user. Thus making the information easier for the user to see. The computer detects which area on a display device has the user's interest by using a gaze-tracking device to monitor where the user looks on the display. After determining which area interests the user the invention expands the display of the information in that area.

Application Filing Date (1):
19960625

Brief Summary Text (8):
Each window displays information generated by an associated application or system program. Further, there may be several windows simultaneously present on the desktop with each containing information generated by a program. A program presents information to the user through each window by drawing or "painting" images, graphics or text within the window. The user can also move a window to a different location on the display screen and change its size and appearance to arrange the desktop in a convenient manner. The user communicates with the program by "pointing at" objects displayed in the window with a cursor controlled by a pointing device and manipulating the objects as desired. In some cases the program requests additional information from the user in response to a manipulation. This request is presented as a "dialog" that allows the user to provide the requested information to the dialog from the keyboard.

Brief Summary Text (17):
Recently, an eyegaze eyetracking system has been developed as described in The Eyegaze Eyetracking System--Unique Example of a Multiple-Use Technology, 4th Annual 1994 IEEE Dual-Use Technologies and Applications Conference, May, 1994. This system comprises a video camera located below a computer display that monitors one of the user's eyes. The device also contains an infrared light emitting diode (LED) located at the center of the camera's lens to maximize the bright-eye affect. Image processing software on the computer computes the user's gazepoint on the display sixty times a second with an accuracy of about a quarter inch.

Brief Summary Text (25):
Prior art WWW browsers, for example, generally attempt to equally allocate bandwidth to all the data transfers directed towards visible views in a window. Although this approach is clearly better then simply sequentially retrieving data for each view, this approach delays retrieving data that is of the most interest to the user because the available channel bandwidth is divided between the data streams supplying data to the views. Thus, the user must wait an additional time because of uninteresting information using bandwidth that could have been applied to the information of interest.

Brief Summary Text (36):

Nevertheless, a computer display must often carry the same amount of information as a newspaper. Thus, mapping the content of a newspaper onto a display screen reduces the size of the type used for the articles to the point where the text of the article is extremely difficult to read. Further, the magnification method used by word processing programs for globally expanding the displayed text does not work well when presenting many articles on a page because magnifying the entire page, and providing a limited view into the page distances the structure of the page from the viewer. Thus, globally expanding the text is incompatible with presenting as many articles as is desired on the page. Further, globally expanding the page also expands the larger title and headline text more than is needed to make this text readable and at a cost of consuming undue display space that could otherwise be used to present additional information. Thus, there is a need for a mechanism that optimizes the text size for a reader while still preserving the structural indications provided by the page layout.

Brief Summary Text (44):
One aspect of the invention is a computer controlled method for displaying a portion of some of a plurality of articles to a computer user. The computer is equipped with a display device and a gaze-tracking device. The gaze-tracking device being capable of determining a gaze position on said display device. The computer controlled method displays portions of articles on the display device. Next the method detects an article of interest by determining an intersection between the gaze position and a portion of one of the articles. Finally, the method expands the displayed portion of the article of interest on the display.

Brief Summary Text (45):
In another aspect of the invention, an information presentation apparatus is disclosed having a central processing unit, memory, a display device and a gaze tracking device capable of determining a gaze position on the display device. The apparatus includes a display mechanism than is configured to display portions of a plurality of articles on the display device. The apparatus also includes a detection mechanism that detects an intersection of a portion of one of the displayed articles with the user's gaze and thus determines an article of interest to the user. Finally, the apparatus includes an expansion mechanism that expands the portion of the article of interest on the display device.

Brief Summary Text (46):
Yet another aspect of the invention discloses an information presentation system. The system includes a gaze-tracking device that determines a gaze position on the display device where the user is looking. The system includes a display mechanism that displays portions of articles on a display device. Additionally, the system includes a detection mechanism that determines an article of interest by detecting an intersection between the gaze position and a portion of a displayed article. Finally, the system includes an expansion mechanism that expands the displayed portion of the article of interest.

Brief Summary Text (47):
A final aspect of the invention discloses a computer program product having computer readable code embodied in a computer usable storage medium. The computer readable code causes a computer to effect a gaze position determination mechanism used to determine a gaze position from gaze coordinates returned from a gaze-tracking device. Further the invention includes code that displays portions of a number of articles on a display device and detects when the gaze position intersects one of these portions. This article containing that intersected portion is the article of interest. Finally, the invention includes code that expands the intersected portion of the article of interest.

Detailed Description Text (7):
Dialog--A specialized window that is used to obtain additional information from the user. A dialog is often used to obtain options and parameters that are computer

dependent. A good example is a print dialog that is evoked by a print menu command. The print dialog allows the user to specify what printer options are to be used for a particular print job. Generally the dialog allows the user to specify specific parameters and then to either affirm or cancel the command that evoked the dialog. If the user cancels the command, the dialog window is removed and the command that evoked the dialog is aborted. If the user confirms the command the user provided information acquired by the dialog is used in the execution of the command that evoked the dialog.

Detailed Description Text (9):
Graphical User Interface (GUI)--A user interface that allows a user to interact with a computer display by pointing at selectable control areas on the display and activating a command or computer operation associated with the selectable control area. GUIs are well known in the art.

Detailed Description Text (15):
Selectable control area--An area on a computer display that is sensitive to activation of a pointing device. On activation of the pointing device over the selectable control area, a command or computer operation associated with the selectable control area is invoked. Most computer systems that provide a Graphical User Interface (GUI) also provide other methods for invoking these commands or computer operations such as keyboard function keys or command lines.

Detailed Description Text (17):
View--An area in a window where information is provided.

Detailed Description Text (24):
FIG. 1 illustrates a computer system referenced to by the general reference character 102, configured to support the invention. The system 102 includes a processor 133 having an Input/Output ("I/O") section 135, a central processing unit ("CPU") 137 and a memory section 139. The I/O section 135 is connected to a keyboard 141, a disk storage unit 143, a network interface 145 to provide access to a network 117, a display unit 147, a pointing device 148, a gaze-tracker device 155, a speaker 157 and a CD-ROM drive unit 149. The CD-ROM unit 149 can read a CD-ROM medium 151 that typically contains a plurality of programs 153 and data. The CD-ROM 149 and the disk storage unit 143 comprising a filestorage mechanism. One skilled in the art will understand that the filestorage mechanism may comprise read only memory, RAM or other storage technology that allows a computer to access data. Such a computer system is capable of executing programmed logic that embodies the invention.

Detailed Description Text (34):
FIG. 5 illustrates how a preferred embodiment of the invention is used in a WWW browser. The browser application displays a window 501 on the display device. The user invokes a URL to present a webpage containing information encoded in HTML in the window 501. In this particular example, the webpage shows a plurality of three areas of text 503, 511 and 517 along with a plurality of areas of images 505, 513 and 519. Each of these images are displayed in a view. Further, these images are constructed from "large data entities". Each large date entity is composed of a large amount of data that is used to define the entity. Examples of large data entities, among others, are high resolution graphical images, sound and video entities. A plurality of shaded areas 507, 509 and 521 of the images 505, 513 and 519 indicate the amount of the image that has been downloaded at some particular point in time prior to completion. If the user is interested in the image labeled as 513, the user will watch that image 513 fill. Thus, a gaze position area 515 intersects the image 513 thus identifying the image 513 as an area of interest. A complementary approach to determining which image 505, 513 or 519 is of interest is to have the text 503, 511 or 517 associated with each image. Thus, when the user reads the text 511 linked (associated) with an image 513 a gaze position 512 is found on the text. Once the user shows interest in the text, the application can

properly assume that the area of interest is the linked image 513.

Detailed Description Text (36):
FIG. 6 illustrates the process used to allocate bandwidth to a data stream supplying data to an area of interest. The process starts at a terminal 601. At a step 603 the process receives the gaze position from a gaze tracking device as described above. Then at a step 605 the gaze position is used to determine what information or view on the display device intersects with the gaze position. Next in a decision block step 607, the process checks whether the displayed information that intersects the gaze position is relevant. Information that is relevant comprises information that is in the process of being transferred to the display device. In other words, a completely loaded image is not relevant and has no bandwidth allocation because there is no need to increase the bandwidth for an image that has been completely transmitted. Contrast this situation where there is zero bandwidth allocated to a yet-to-be-loaded image. Here the yet-to-be-loaded image has a bandwidth allocation, but the current allocation is zero whereas there is no bandwidth allocation at all for an image that has been completely downloaded. Text is another example of displayed information that usually is not relevant because the user cannot read the text as fast as it is transmitted even with limited bandwidth.

Detailed Description Text (51):
A page from an electronic newspaper is illustrated in FIG. 10. This information is displayed on a computer display screen generally in a window as discussed above. Here a window 1001 contains a headline 1003, a plurality of four article titles 1005, 1009, 1013 and 1017 and a partial text 1007, 1011, 1015, and 1019 of the titled articles. A computer for displaying the electronic newspaper that is equipped with a gaze tracking device can rely on the normal reading pattern of a user. That is, that the user first reads the major headline 1003, then the article title 1005, 1009, 1013 or 1017 followed by the article itself 1007, 1011, 1015, or 1019. Thus, when the gaze tracker indicates that the user is reading an article title, the invention starts expanding the article text by a magnification factor to an optimal size for the user.

Detailed Description Text (55):
FIG. 12 illustrates another approach to managing the information displayed in a window 1201. Here, a title 1205 and text of an article AA 1207 is expanded and simply overlays a plurality of other articles 1211, 1215, and 1219 in the window 1201. Because the other articles and titles are obscured, the text is reduced when the user's gaze leaves the window 1201 or when the user invokes a command function either by use of a pointing device, a command line or other command invocation method. Because the magnified view overlays the other views, a preferred embodiment places the magnified view within a boarder 1217 to separate the magnified view from the other views.

Detailed Description Text (62):
FIG. 16 illustrates a possible second page of information. Again, the information is provided within views contained in a window 1601. Now a plurality of articles 1607, 1611, 1615 and 1619 are all scientific or technology based, but with different levels of difficulty extending from articles of interest to the lay reader to those that are directed toward the advanced elemental particle physicist. Further, both a Major Scientific Headline 1603 and an advertising 1621 can be selected to be of interest to the user. This allows the information provider to narrowly target advertising and articles to each user. Again the information provider can continue to refine and narrow the selection of information presented to the user on subsequent pages depending on the interest shown in a plurality of article titles 1605, 1609, 1613, 1617, the time spent with reading each article 1607, 1611, 1615 and 1619 and the time spent looking at the advertisement 1621 of the current page.

Other Reference Publication (2):
"Noncommand USer Interfaces", by Jakob Nielsen, Communications of the ACM, Apr. 1993, vol. 36, No. 4, pp. 83-99.

CLAIMS:

1. A computer controlled method for displaying a portion of some of a plurality of articles to a user of a computer; said computer having a display device, and a gaze-tracking device; said gaze-tracking device determining a gaze position on said display device; said plurality of articles including a first article; said computer controlled method comprising the steps of:

(a) displaying said portion of some of said plurality of articles on said display device;

(b) detecting an article-of-interest by determining an intersection between said gaze position and said displayed portion of said first article;

(c) expanding said portion of said article-of-interest resulting in an expanded article-of-interest; and

(d) displaying said expanded article-of-interest.

10. An information presentation apparatus configured to display a portion of some of a plurality of articles; said information presentation apparatus having a central processing unit (CPU), a memory, a display device, and a gaze-tracking device; said gaze-tracking device capable of determining a gaze position on said display device; said plurality of articles including a first article; said information presentation apparatus comprising:

a display mechanism configured to display said portion of some of said plurality of articles on said display device;

a detection mechanism configured to detect an article-of-interest by determining an intersection between a gaze position and a portion of said first article displayed by said display mechanism on said display device; and

an expansion mechanism configured to expand said portion of said article-of-interest resulting in an expanded article-of-interest displayed on said display device.

19. An information presentation system configured to display a portion of some of a plurality of articles; said information presentation system having a display device and a gaze-tracking device; said gaze-tracking device for determining a gaze position on said display device;. said plurality of articles including a first article; said information presentation system comprising:

a display mechanism configured to display said portion of some of said plurality of articles on said display device;

a detection mechanism configured to detect an article-of-interest by determining an intersection between a gaze position and a portion of said first article displayed by said display mechanism on said display device; and

an expansion mechanism configured to expand said portion of said article-of-interest resulting in an expanded article-of-interest displayed on said display device.

28. A computer program product comprising:

a computer usable storage medium having computer readable code embodied therein for causing a computer to display a portion of some of a plurality of articles on a display device; said plurality of articles including a first article; said computer readable code comprising:

computer readable code devices configured to cause said computer to effect a gaze position determination mechanism configured to determine a gaze position from gaze coordinates returned from a gaze-tracking device;

computer readable code devices configured to cause said computer to effect a display mechanism configured to display said portion of some of said plurality of articles on said display device;

computer readable code devices configured to cause said computer to effect a detection mechanism configured to detect an article-of-interest by determining an intersection between said gaze position and said portion of said first article displayed by said display mechanism on said display device; and

computer readable code devices configured to cause said computer to effect an expansion mechanism configured to expand said portion of said article-of-interest resulting in an expanded article-of-interest displayed on said display device.